

Facultad: Ingeniería

Escuela: Electrónica

Asignatura: Interfaces y Periféricos

Tema: Manejo de pantallas LCD, botones y teclados

Objetivos Específicos

1. Realizar Interfaces con Arduino haciendo uso de dispositivos de entrada como botones y teclados.
2. Realizar Interfaces con Arduino haciendo uso de dispositivos de salida como Pantalla LCD.
3. Desarrollar aplicaciones mediante sketches

Material y Equipo

No	Cantidad	Descripción
1	1	Una computadora con sistema operativo Windows y con el programa Arduino IDE instalados
2	1	Cable USB (tipo A hacia tipo B)
3	1	Tarjeta Arduino UNO R2 o R3
4	3	Resistencias de 4.7k Ω
5	3	Pulsadores
6	1	Shield LCD (16x2)
7	1	Teclado matricial de 4x4
8	1	Breadboard
9	1	Pinza
10	1	Cortadora
11	varios	Cables UTP

Introducción Teórica

Phi_interfaces es la primera y única librería de código abierto Arduino que maneja todos los dispositivos de entrada para Arduino, tales como pulsadores, teclados matriciales, codificadores rotatorios, botones analógicos, incluso la comunicación por medio de Bluetooth con teléfonos inteligentes, todo ello bajo un marco común. Vea la figura 1 para tener una idea del concepto.

Por el momento la librería contiene clases para los botones, teclados matriciales, botones analógicos, codificadores rotatorios, pantallas de teclados LCD seriales (o entradas de teléfonos inteligentes o simuladas a través de teclados serie), además existen formas muy fáciles de ampliar la interfaz para incluir otros dispositivos como teclados capacitivos,

controles remoto IR, ratón PS / 2 teclados, teclados de pantalla táctil, escudos Ethernet, etc. Con esta librería, ya no tienen que buscar y aprender muchas librerías, es decir una para cada tipo de insumo. Sólo tiene que insertar esta librería y podrá utilizar cualquiera de todos los dispositivos de entrada antes mencionado.



Figura 1. Librería Phi_interfaces.

La librería Phi_interfaces es de mucha ayuda para la construcción de diversas aplicaciones, donde se necesite la utilización de los dispositivos de entrada antes mencionados.

Debido a que es una librería ajena al fabricante de Arduino, debe ser descargada de la página del creador y agregada a la carpeta donde se encuentra alojadas todas las librerías en nuestra computadora. Para ello hay que seguir el siguiente procedimiento:

1) Descargar la librería desde la página del creador.

http://liudr.wordpress.com/2012/01/25/phi_interfaces-library-1-0-released/

2) La librería vendrá comprimida, por lo que hay que descomprimirla para poder utilizarla.

El archivo descomprimido es una carpeta con el nombre phi_interfaces, esta carpeta debe ser copiada dentro de la carpeta de las librerías de Arduino IDE, la cual se encuentra dentro de la carpeta que usted accede para poder lanzar el Arduino IDE, su ubicación exacta dependerá de donde usted la ubico después de haberlo descargado ya que recuerde que es un programa portable (es decir no se necesita instalar para ser utilizado). (Ejemplo: C:\Documents and Settings\Administrador\Mis documentos \arduino-1.0.1-windows\arduino-1.0.1\libraries).

Para su uso con Linux, deberá copiar (como super-usuario o como administrador) en la carpeta /usr/share/arduino/libraries y después dar permisos de archivo para lectura escritura al usuario de sistema, a root y a otros. Para cualquier consulta acerca de este procedimiento consulte con un manual de usuario de linux según la distribución que esté utilizando.

Es decir que la carpeta de interés donde necesitamos alojar la librería descargada (phi_interfaces) es la carpeta de librerías (libraries).

Con esta acción la librería ya podrá ser incluida y utilizada la librería en cualquier sketch simplemente agregando al inicio del sketch la línea de código `#include <phi_interfaces.h>`.

Para el uso de dispositivos de salida como una pantalla LCD no es necesaria la descarga de ninguna librería extra, ya que el fabricante ha incluido una librería para el control de LCD, y para hacer uso de ella basta con agregar al inicio del sketch la línea de código `#include`

<LiquidCrystal.h>. Esto es en caso se posea una LCD de 2 líneas por 16 columnas suelta

Para el caso del Shield¹ de arduino presente en el laboratorio este contiene una pantalla LCD Hitachi HD44780U de 2 filas por 16 columnas y 5 botones adjuntos a la entrada analógica A0



Figura 2. Shield de pantalla LCD para Arduino

La conexión de los pines al núcleo de Arduino se resume en la siguiente tabla

PIN	Función
Analog 0	Botones (select, up, right, down, y left)
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS (Datos o Señal de selección digital)
Digital 9	Habilitador (Enable)
Digital 10	Control de luz trasera (Backlit Control)

Tabla 1. Conexiones de los pines del Shield LCD al núcleo Arduino

¹ Un Shield para arduino es una tarjeta que se conecta a los pines del núcleo y le brinda otras funcionalidades, por ejemplo, pantalla LCD, comunicación Ethernet, control de motores DC y Paso entre otros.

Procedimiento

Parte I – Botones Pulsadores.

1. Inicie el entorno de desarrollo de Arduino y dentro de este digite y guarde el siguiente código.

```
// Sketch para el control de botones pulsadores
#include <phi_interfaces.h> //Incluyendo librería para control de teclados.
#define btn_a 2           //definiendo los pines en que estarán...
#define btn_b 3           //... conectados los botones
#define btn_c 4
#define total_buttons 3   // definiendo cantidad de botones a utilizar

char mapping[]={'A','B','C'}; //Esta es una lista con los nombres de cada botón
byte pins[]={btn_a,btn_b,btn_c}; //Pines conectados a los 3 botones
phi_button_groups my_btns(mapping, pins, total_buttons);

void setup(){
  Serial.begin(9600); // configurando velocidad de transmisión de puerto serial
}

void loop(){
  char temp;           //declarando variable para almacenar el valor del botón
  temp=my_btns.getKey();// Usando objeto phi_button_groups para acceder al grupo de botones
  if (temp!=NO_KEY) Serial.write(temp); // Enviando valor de botón presionado al puerto serial
}
```

Listado 1. Sketch para control de botones pulsadores.

2. Arme el circuito que se muestra a continuación, dicho circuito servirá para comprobar el funcionamiento del Sketch codificado en el numeral anterior.

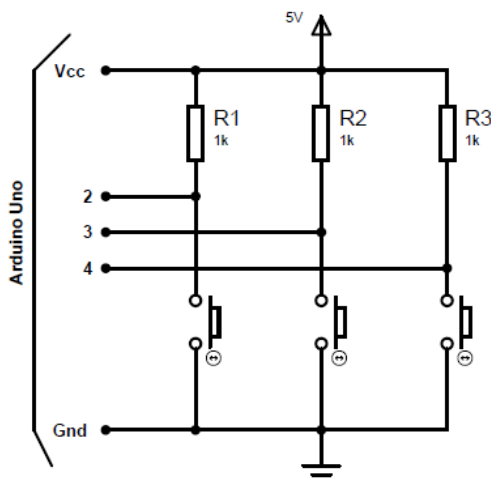


Figura 3. Diagrama de circuito para la parte I (Botones).

3. Conecte el Arduino a la PC y cargue el sketch codificado en el primer numeral, compruebe su funcionamiento, para ello deberá abrir el monitor serial desde el entorno de desarrollo Arduino y observe lo que sucede en este cuando es presionado alguno de los pulsadores.
4. Notifique a su instructor que ha terminado esta parte para su correspondiente revisión y para continuar con la siguiente parte.

Parte II – Teclado Matricial

1. Cree un nuevo archivo desde el entorno de desarrollo Arduino, digite y guarde el siguiente código.

```
// Sketch para el control de teclados matriciales
#include <phi_interfaces.h> //Incluyendo librería para control de teclados.
#define btns_per_column 4 //definiendo cantidad de botones por columna
#define btns_per_row 4 //definiendo cantidad de botones por fila
char mapping[]={'1','2','3','A','4','5','6','B','7','8','9','C','*','0','#','D'}; //Esta es una lista con los...
                                                                    //...nombres de
cada botón
byte pins[]={2, 3, 4, 5, 6, 7, 8, 9}; // pines de fila después pines de columna
phi_matrix_keypads panel_keypad(mapping, pins, btns_per_row, btns_per_column);

void setup()
{
  Serial.begin(9600); // configurando velocidad de transmisión de puerto serial
}

void loop()
{
  char temp; //declarando variable para almacenar el valor del botón
  temp=panel_keypad.getKey(); //Usando objeto phi_keypads para acceder al teclado
  if (temp!=NO_KEY) Serial.write(temp); // Enviando valor de botón presionado al puerto serial
}
```

Listado 2. Sketch para control de teclado matricial

2. Arme el circuito que se muestra a continuación, dicho circuito servirá para comprobar el funcionamiento del Sketch codificado en el numeral anterior.

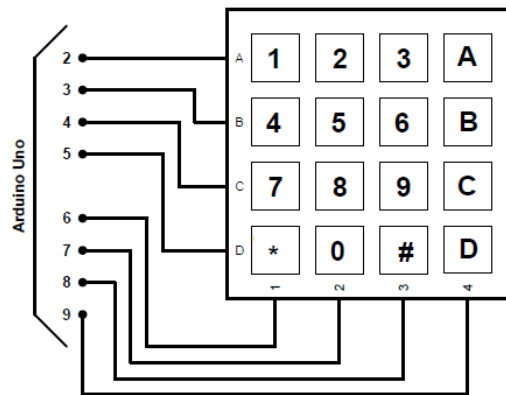


Figura 4. Diagrama de circuito para la parte II (Teclado Matricial).

3. Cargue el sketch codificado en el primer numeral, compruebe su funcionamiento, para ello deberá abrir el monitor serial desde el entorno de desarrollo Arduino y observe lo que sucede en este cuando es presionado alguno de los botones del teclado matricial.
4. Notifique a su instructor que ha terminado esta parte para su correspondiente revisión y para continuar con la siguiente parte.

Parte III – Pantallas de Cristal Líquido (LCD)

1. Cree un nuevo archivo desde el entorno de desarrollo Arduino, digite y guarde el siguiente código.

```
#include <LiquidCrystal.h>
#include <phi_interfaces.h>

#define buttons_per_column 5 // El pin analógico tiene cinco botones con resistencias.
#define buttons_per_row 1 // Hay un pin analógico en uso

byte keypad_type=Analog_keypad;
char mapping[]={'1','2','3','4','5'}; // un teclado analógico
byte pins[]={0}; // Los números de pin son de entradas analógicas.
int values[]={0, 99, 257, 410, 640}; //Se necesitan para incrementarse de forma monótona.
// Estos valores son los que se miden en la entrada analógica al presionarse los botones
phi_analog_keypads panel_keypad(mapping, pins, values, buttons_per_row, buttons_per_column);
multiple_button_input* pad1=&panel_keypad;
LiquidCrystal lcd(8, 9, 4, 5, 6, 7); //Distribución de pines de pantalla con Arduino

void setup()
{
  lcd.begin(16, 2); // Inicializa la librería
  lcd.setCursor(0,0); //Ubica el cursor en la fila 0, columna 0
  lcd.print("Presione botones"); // Imprime un mensaje simple
}

void loop()
{
  lcd.setCursor(9,1); // Mueve el cursor a Fila 1 y columna 9
```

```

lcd.print(millis()/1000);    // Despliega los segundos transcurridos desde el encendido.
lcd.setCursor(0,1);
byte temp=panel_keypad.getKey(); // Usa el objeto phi_keypads para acceder al teclado.

if (temp!=NO_KEY)

switch (temp) // Dependiendo del botón presionado se ejecuta la acción.
{
  case '1':
  {
    lcd.print("RIGHT ");
    break;
  }
  case '2':
  {
    lcd.print("UP   ");
    break;
  }
  case '3':
  {
    lcd.print("DOWN  ");
    break;
  }
  case '4':
  {
    lcd.print("LEFT   ");
    break;
  }
  case '5':
  {
    lcd.print("SELECT ");
    break;
  }
}
delay(10);
}

```

Listado 3. Uso de botones integrados en el Shield LCD con la librería phi_interfaces

2. Conecte el LCD Shield a los pines del Arduino, tenga cuidado que ningún pin quede fuera de los conectores.
3. Conecte el cable USB al Arduino y descargue el programa y compruebe su funcionamiento. Nota: El botón RST del Shield reinicia al Arduino y su programa.
4. Después de haber comprobado el programa haga un nuevo Sketch y copie el siguiente programa:

```

//Sketch para el control de LCD
// Incluyendo librerías a utilizar:

#include <LiquidCrystal.h>
#include <phi_interfaces.h>
#define buttons_per_column 5 // El pin analógico tiene cinco botones con resistencias.
#define buttons_per_row 1 // Hay un pin analógico en uso

```

```

byte keypad_type=Analog_keypad;
char mapping[]={'1','2','3','4','5'}; // un teclado analógico
byte pins[]={0}; // Los numeros de pin son de entradas analógicas.
int values[]={0, 99, 257, 410, 640}; //Se necesitan para incrementarse de forma monótona.
// Estos valores son los que se miden en la entrada analogica al presionarse los botones
phi_analog_keypads panel_keypad(mapping, pins, values, buttons_per_row, buttons_per_column);
multiple_button_input* pad1=&panel_keypad;

/* Configuración de LCD
RS Enable R/W D4 D5 D6 D7 VSS VDD Vo
8 9 Gnd 4 5 6 7 Gnd Vcc Pot
*/
int BacklitPin = 10;
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
int conteo = 0; //declarando e inicializando variable
void setup()
{
    pinMode(BacklitPin, OUTPUT);
    digitalWrite(BacklitPin,HIGH);
    lcd.begin(16, 2); // Configurando el numero columnas y filas de LCD
    lcd.setCursor(3, 0); //coloca el cursor en columna 3, fila 0
    lcd.print("BIENVENIDO"); // Imprime mensaje en LCD
    lcd.setCursor(4, 1); //coloca el cursor en columna 4, fila 1
    lcd.print("IYP-UDB"); // Imprime mensaje en LCD
    delay(2500); // retardo antes de pasar a Loop
    lcd.clear(); //limpia la pantalla
}
void loop()
{
    char temp; //declarando variable para almacenar el valor del botón
    temp=panel_keypad.getKey(); // Usando objeto phi_button_groups para acceder al grupo de botones
    if (temp=='4') //Si es LEFT
    { digitalWrite(BacklitPin, HIGH); //Activa la luz trasera
    }
    if (temp=='5') //Si es SELECT
    {
        digitalWrite(BacklitPin, LOW); //Desactiva la luz trasera
    }
    if (temp=='3') //Si es DOWN
    {
        {
            conteo++;
            lcd.clear();
        }
        if (conteo < 26)
        {
            lcd.setCursor(0, 0); //coloca el cursor en columna 0, fila 0
            lcd.print(conteo); // Imprime mensaje en LCD
        }
        else
        {
            if (conteo == 26)
            {
                lcd.clear(); // Limpiar LCD
                lcd.setCursor(6, 1); //coloca el cursor en columna 6, fila 1
            }
        }
    }
}

```

```
lcd.print(conteo); // Imprime mensaje en LCD2
delay(15);
}
else
{
  lcd.setCursor(6, 1); //coloca el cursor en columna 6, fila 1
  lcd.print(conteo); // Imprime mensaje en LCD
}
if (conteo >= 51)
{
  lcd.clear();
  lcd.setCursor(4, 0); //coloca el cursor en columna 4, fila 0
  lcd.print("***Fin***"); // Imprime mensaje en LCD
  delay(15);
  conteo=0;
}
}
}
```

Listado 4. Sketch para control de LCD

5. Cargue el sketch codificado en el anterior listado, compruebe su funcionamiento, para ello deberá presionar varias veces el pulsador DOWN del Shield, observe lo que sucede en la LCD, también presione SELECT y LEFT y cuál es su efecto.
6. Notifique a su instructor que ha terminado esta parte para su correspondiente revisión y evaluación final.

Análisis de Resultados

- ¿Qué función realiza la instrucción `lcd.setCursor(a, b)`?
- ¿Cuál es la función del que realiza Pin VO en el LCD?
- ¿Se pueden agregar más botones a otras entradas analógicas?

Investigación Complementaria

- Investigue cómo se puede hacer lectura de botones de forma serial utilizando la librería `Phi_interfaces`.
- El Shield tiene integrados 5 botones que son leídos desde una sola entrada analógica, investigue cómo es la conexión del circuito necesario para hacer esto.

Bibliografía

- Referencia en línea del lenguaje: <http://arduino.cc/en/Reference/HomePage>
- Massimo Banzi, "Getting started with Arduino" O'Reilly Media / Make Publisher, Marzo 2009, ISBN: 978-0-596-15551-3
- Creador de Librería `Phi_interfaces`:
http://liudr.wordpress.com/2012/01/25/phi_interfaces-library-1-0-released/

Guía 4: Manejo de Pantallas y Teclados

Alumno:

Maquina No:

Docente:

GL:

Fecha:

EVALUACION					
	%	1-4	5-7	8-10	Nota
CONOCIMIENTO	25%	Conocimiento deficiente de los fundamentos teóricos	Conocimiento y explicación incompleta de los fundamentos teóricos	Conocimiento completo y explicación clara de los fundamentos teóricos	
APLICACIÓN DEL CONOCIMIENTO	70%	No Terminó completamente el programa y comentó mal el código (30%) No realizó el circuito correctamente (40%)	Terminó completamente el programa pero los comentarios no estaban del todo correctos El circuito funcionó	Terminó completamente el programa con sus comentarios correctos El circuito funcionó	
ACTITUD	2.5 %	Es un observador pasivo.	Participa ocasionalmente o lo hace constantemente pero sin coordinarse con su compañero.	Participa propositiva e integralmente en toda la práctica.	
	2.5 %	Es ordenado; pero no hace un uso adecuado de los recursos	Hace un uso adecuado de los recursos, respeta las pautas de seguridad; pero es desordenado.	Hace un manejo responsable y adecuado de los recursos conforme a pautas de seguridad e higiene.	
TOTAL	100 %				